Genetic program based data mining of fuzzy decision trees and methods of improving convergence and reducing bloat

James F. Smith III*, ThanhVu H. Nguyen

Naval Research Laboratory, Code 5741, Washington, D.C., 20375-5000

ABSTRACT

A data mining procedure for automatic determination of fuzzy decision tree structure using a genetic program (GP) is discussed. A GP is an algorithm that evolves other algorithms or mathematical expressions. Innovative methods for accelerating convergence of the data mining procedure and reducing bloat are given. In genetic programming, bloat refers to excessive tree growth. It has been observed that the trees in the evolving GP population will grow by a factor of three every 50 generations. When evolving mathematical expressions much of the bloat is due to the expressions not being in algebraically simplest form. So a bloat reduction method based on automated computer algebra has been introduced. The effectiveness of this procedure is discussed. Also, rules based on fuzzy logic have been introduced into the GP to accelerate convergence, reduce bloat and produce a solution more readily understood by the human user. These rules are discussed as well as other techniques for convergence improvement and bloat control. Comparisons between trees created using a genetic program and those constructed solely by interviewing experts are made. A new co-evolutionary method that improves the control logic evolved by the GP by having a genetic algorithm evolve pathological scenarios is discussed. The effect on the control logic is considered. Finally, additional methods that have been used to validate the data mining algorithm are referenced.

Keywords: data mining, knowledge discovery, fuzzy logic, genetic program, genetic algorithm, co-evolution

1. INTRODUCTION

Two fuzzy logic based resource managers (RMs) have been developed that automatically allocate resources in real-time[1-3]. Both RMs were evolved by genetic programs (GPs). The GPs were used as data mining functions. Both RMs have been subjected to a significant number of verification experiments.

The most recently developed RM is the main subject of this paper. This RM automatically allocates unmanned aerial vehicles (UAVs) that will ultimately measure atmospheric properties in a cooperative fashion without human intervention[2,3]. This RM will be referred to as the UAVRM. It consists of a pre-mission planning algorithm and a real-time control algorithm that runs on each UAV during the mission allowing the UAVs to automatically cooperate.

The previous RM was evolved to control electronic attack functions distributed over many platforms[1]. It will be referred to as the electronic attack RM (EARM).

New approaches for improving the convergence of the genetic program that evolve control and planning logic are discussed. Such procedures involve the use of symbolic algebraic techniques not previously explored, a terminal set that includes fuzzy concepts and their complements, the use of fuzzy rules, etc. Two distinct examples of data mining fuzzy decision trees as well as their subtrees are provided in detail. A co-evolutionary data mining procedure for improving the results is introduced. Experiments to validate the evolved algorithms are referenced.

The particular approach to fuzzy logic used by the UAVRM is the fuzzy decision tree[1-4]. The fuzzy decision tree is an extension of the classical artificial intelligence concept of decision trees. The nodes of the tree of degree one, the leaf nodes are labeled with what are referred to as root concepts. Nodes of degree greater than unity are labeled with composite concepts, i.e., concepts constructed from the root concepts[4-7] using logical connectives and modifiers. Each

* Correspondence: Email: jfsmith@drsews.nrl.navy.mil

Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2007, edited by Belur V. Dasarathy, Proc. of SPIE
Vol. 6570, 65700A, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.716973

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Genetic program based data mining of fuzzy decision trees and methods of improving convergence and reducing bloat	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Code 5741,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A data mining procedure for automatic determination of fuzzy decision tree structure using a genetic program (GP) is discussed. A GP is an algorithm that evolves other algorithms or mathematical expressions. Innovative methods for accelerating convergence of the data mining procedure and reducing bloat are given. In genetic programming, bloat refers to excessive tree growth. It has been observed that the trees in the evolving GP population will grow by a factor of three every 50 generations. When evolving mathematical expressions much of the bloat is due to the expressions not being in algebraically simplest form. So a bloat reduction method based on automated computer algebra has been introduced. The effectiveness of this procedure is discussed. Also, rules based on fuzzy logic have been introduced into the GP to accelerate convergence, reduce bloat and produce a solution more readily understood by the human user. These rules are discussed as well as other techniques for convergence improvement and bloat control. Comparisons between trees created using a genetic program and those constructed solely by interviewing experts are made. A new co-evolutionary method that improves the control logic evolved by the GP by having a genetic algorithm evolve pathological scenarios is discussed. The effect on the control logic is considered. Finally, additional methods that have been used to validate the data mining algorithm are referenced.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

root concept has a fuzzy membership function assigned to it. Each root concept membership function has parameters to be determined. For the UAVRM, the parameters were set based on expertise.

The UAVRM consists of three fuzzy decision trees. The use of genetic program based data mining (DM) to create two of the trees is discussed in this paper. The first tree discussed is the AUP fuzzy decision tree. The acronym AUP refers to the tree's function: it is used to assign UAVs to paths (AUP). The second tree described is the PH fuzzy decision tree. This tree determines a UAVs priority of helping (PH) another UAV that makes a request. Both of the fuzzy decision trees (FDTs) make use of the risk tree which is discussed in the literature[2, 3].

Data mining is the efficient extraction of valuable non-obvious information embedded in a large quantity of data[8]. Data mining consists of three steps: the construction of a database that represents truth; the calling of the data mining function to extract the valuable information, e.g., a clustering algorithm, neural net, genetic algorithm, genetic program, etc; and finally determining the value of the information extracted in the second step, this generally involves visualization.

In a previous paper a genetic algorithm (GA) was used as a data mining function to determine parameters for fuzzy membership functions[7]. Here, a different data mining function, a genetic program[9] is used. A genetic program is a problem independent method for automatically evolving computer programs or mathematical expressions.

The GP data mines fuzzy decision tree structure, i.e., how vertices and edges are connected and labeled in a fuzzy decision tree. The GP mines the information from a database consisting of scenarios.

Section 2 describes the UAVRM's AUP FDT that assigns UAVs to paths. Section 3 introduces the FDT that allows UAVs to automatically cooperate, the PH FDT. Section 4 develops the formalism and methodology for the genetic program based data mining used to evolve the AUP tree. Section 5 provides details as to how the PH tree was created using GP based data mining. Section 6 gives an overview of a recently invented co-evolutionary data mining procedure where a GP mines a scenario data base (DB) to create logic and a genetic algorithm is subsequently used to create scenarios that show flaws in the GP built logic. These scenarios then become part of the GP's DB. The GP mines the new DB to create logic superior to that originally created with the older more limited GP DB. In section 7 a short discussion of computational experiments for validating the FDTs is given. Finally, section 8 provides a summary.

2. UAV PATH ASSIGNMENT ALGORITHM, THE AUP TREE

Knowledge of meteorological properties is fundamental to many decision processes. The UAVRM enables a team of UAVs to cooperate and support each other as they measure atmospheric meteorological properties in real-time. Each UAV has onboard its own fuzzy logic based real-time control algorithm. The control algorithm renders each UAV fully autonomous; no human intervention is necessary. The control algorithm aboard each UAV will allow it to determine its own course, change course to avoid danger, sample phenomena of interest that were not preplanned, and cooperate with other UAVs.

The UAVRM determines the minimum number of UAVs required for the sampling mission. It also determines which points are to be sampled and which UAVs will do the sampling. To do this, both in the planning and control stages it must solve an optimization problem to determine the various paths that must be flown. Once these paths are determined the UAVRM uses the AUP fuzzy decision tree to assign UAVs to the paths.

The AUP fuzzy decision tree is displayed in Figure 1. The various fuzzy root concepts make up the leaves of the tree, i.e., those vertices of degree one. The vertices of degree higher than one are composite concepts.

Starting from the bottom left of the VMR subtree in Figure 1 and moving to the right, the fuzzy concepts "RISK-TOL," "VALUE," "LOW RISK," and "FAST," are encountered. These four fuzzy root concepts are combined through logical connectives to give the composite concept "VMR." Although four concepts are now used to construct VMR it originally only used the concepts related to value and mission risk, and was called the Value-Mission-Risk (VMR) subtree.
These concepts are developed in greater mathematical detail in the literature[2,3]. The fuzzy concept "RISK-TOL" refers to an individual UAV's risk tolerance. This is a number assigned by an expert indicating the degree of risk the UAV

may tolerate. A low value near zero implies little risk tolerance, whereas, a high value near one implies the UAV can be subjected to significant risk.

The concept "VALUE" is a number between zero and one indicating the relative value of a UAV as measured against the other UAVs flying the mission. The concept "VALUE" changes from mission to mission depending on which UAVs are flying.

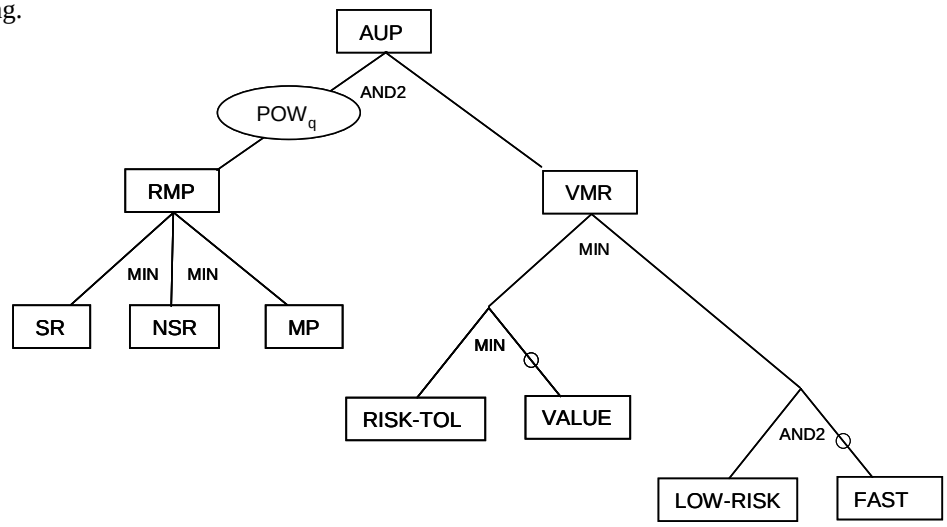

Figure 1: AUP tree evolved through genetic program based data mining.

The concept "FAST" relates to how fast the UAV is and builds in measures of the UAV's reliability estimates as well as its risk tolerance and the mission's priority.

The concept "LOW RISK" quantifies experts' opinions about how risky the mission is. It takes a value of one for low risk missions and a value near zero for high risk missions.

Each vertex of the "VMR" tree uses a form of "AND" as a logical connective. In fuzzy logic, logical connectives can have more than one mathematical form. Based on expertise it was useful to allow two types of ANDs to be used. The two mathematical forms of AND used are the "MIN" operator and the algebraic product denoted in Figure 1 as "AND2." When a "MIN" appears on a vertex then the resulting composite concept arises from taking the minimum between the two root concepts connected by the "MIN." When an "AND2" appears it means that the resulting composite concept is the product of the fuzzy membership functions for the two concepts connected by the AND2.

The final subtree of AUP that needs to be described is the reliability-mission priority (RMP) subtree. RMP consists of a "MIN" operation between three fuzzy concepts. These concepts are "SR" which refers to an expert's estimate of the sensor reliability, "NSR" which refers to an expert's estimate of the non-sensor system reliability and "MP" a fuzzy concept expressing the mission's priority.

The RMP tree is modified by the fuzzy logic modifier "POW_q," that appears on the AUP tree. The modifier, "POW_q," raises the output of the RMP tree to the power "q." For GP based data mining a value of two was found for "q," resulting in the output of the RMP tree being squared.

The AUP tree is observed to consist of the VMR subtree and the "POW_q," modified RMP subtree with AND2 logical connectives at each vertex. These fuzzy concepts and their related fuzzy membership functions, as well as additional information are given in much greater detail in the literature[2, 3].

The AUP tree given in Figure 1 was originally created using human expertise alone. The rediscovery of this tree using GP based data mining is described in section 4.

3. FUZZY DECISION TREE FOR PROVIDING HELP

The next fuzzy decision tree to be developed is the "priority of helping"(PH) decision tree. This tree allows the UAVs to determine how they should support each in real-time as the need arises. When a UAV requires help in making a measurement, its diagnostic systems indicate a sensor might be malfunctioning or there is a clear indication of a malfunction, a UAV can request that another UAV provide help. The request for help is sent out as an omni-directional message. When a UAV sends out an omni-directional request for support, those UAVs receiving the message will calculate their fuzzy priority of providing help, denoted as μ_{PH}. The UAV that will ultimately help the requester is the one with the highest value of μ_{PH}. The fuzzy concept, priority of helping, takes into account properties of the potential supporter. The set of UAVs that receive the request for help from UAV(i) at time t is denoted as $help(i,t)$. If UAV(i) requests help at time t and UAV(j) receives the message then UAV(j) will take into account the necessary travel time it will consume in helping UAV(i), as well as the relative amounts of fuel and battery life the potential helper, UAV(j), has at the time the request is received. Define the relative degree of fuel and battery power left at time, t, that UAV(j) might use to help UAV(i) as

$$\mu_{fuel}(i,j,t) = \frac{fuel(UAV(j),t)}{\max\limits_{k \in help(i,t)} fuel(UAV(k),t)} \tag{1}$$

and

$$\mu_{battery}(i,j,t) = \frac{battery(UAV(j),t)}{\max\limits_{k \in help(i,t)} battery(UAV(k),t)} \tag{2}$$

Define the relative degree of UAV(j)'s "not-separation" from UAV(i) as

$$\mu_{nsep}(i,j,t) = \min\left[1, \beta + 1 - \frac{T(UAV(j), Q_{req}(j,i))}{\max\limits_{k \in help(i,t)} T(UAV(k), Q_{req}(j,i))}\right] \tag{3}$$

where $T(UAV(j), Q_{req}(j,i))$ is the time it would take UAV(j) to travel the path $Q_{req}(j,i)$ from the point $\vec{pos}(j,t)$ where UAV(j) receives request for help at time, t, from UAV(i) to the final point , $\vec{q}_{n_{request}(i),i}$, where it would start helping UAV(i). The subscript *"req"* on $Q_{req}(j,i)$ is an abbreviation for "requested path." The quantity, $n_{request}(i)-1$, is the number of points that UAV(j) would pass through in going from its position at time, *t*, to the first new sample point, $\vec{q}_{n_{request}(i),i}$.

The travel time $T(UAV(j), Q_{req}(j,i))$ is determined by an A* algorithm[2, 3]. It includes sampling and non-sampling velocities.

The quantity, β, is added so that $\mu_{nsep}(i,j,t)$ remains nonzero even for the UAV in the set $help(i,t)$ that will take the maximum amount of time, which leaves open the possibility of the slowest UAV participating in the coordinated team. The quantity, β, is an additive constant to be determined such that

$$0 < \beta \le 1 . \tag{4}$$

Let the path from $\vec{q}_{n_{request}(i)+1,i}$ the first flight point beyond $\vec{q}_{n_{request}(i),i}$ to $\vec{P}_{base}$, the position of the base that UAV(j)

returns to after helping UAV(i) be denoted as $Q_{sar}(j,i)$, where the subscript "*sar*" denotes "sample and return." The full path that UAV(j) will fly in support of UAV(i) is denoted as

$$SPath(j,i) \equiv \left[Q_{req}(j,i), Q_{sar}(j,i)\right] \tag{5}$$

where the notation *SPath* is an abbreviation for "support path." It should be recalled that $Q_{req}(j,i)$ is a matrix of order $\left(1+n_{request}(i)\right)\times 3$ where the "3" arises from representing points in three spatial dimensions. If the path $Q_{sar}(j,i)$ has $n_{sar}(j,i)$ points then $Q_{sar}(j,i)$ is a $n_{sar}(j,i)\times 3$ matrix. The path $SPath(j,i)$ is then represented by a $\left(1+n_{request}(i)+n_{sar}(j,i)\right)\times 3$ matrix. The path $Q_{sar}(j,i)$ and subsequently $SPath(j,i)$ can contain non-sampling points, new sampling points contributed by UAV(i) and old sampling points originally assigned to UAV(j), assuming UAV(j) has enough fuel and battery time left to sample all these points.

As an intermediate step define the quantity below

$$\begin{aligned} &FB(UAV(j), SPath(j,i)) \equiv \\ &\quad \chi\left(\min\left[fuel(UAV(j),t)+\varepsilon_{fuel}, battery(UAV(i),t)+\varepsilon_{battery}\right]-T(UAV(j), SPath(j,i))\right) \end{aligned} \tag{6}$$

The parameters ε_{fuel} and $\varepsilon_{battery}$ are added to make sure that UAV(j) has sufficient fuel and battery time in the face of travel uncertainties such as head winds which may prolong flight times. The notation, "FB," in the name of the function in (6) is an abbreviation for "fuel and battery."

Let UAV(j)'s fuzzy degree of membership in the fuzzy concept "fuel-battery-separation" (FBS) be defined as

$$\mu_{FBS}(UAV(j),UAV(i),SPath(j,i)) \equiv FB(UAV(j),SPath(j,i))\cdot \mu_{nsep}(i,j,t)\cdot \mu_{fuel}(i,j,t)\cdot \mu_{battery}(i,j,t) \tag{7}$$

The FBS fuzzy decision tree is depicted in Figure 2.

Finally, enough formalism has been developed to define the membership function for the fuzzy concept "priority of helping" (PH) for UAV(j) to help UAV(i). This membership function is defined as

$$\begin{aligned} &\mu_{PH}(UAV(j),UAV(i),SPath(j,i)) \equiv \\ &\quad \min\left[\mu_{FBS}(UAV(j),UAV(i),SPath(j,i)), \mu_{AUP}(UAV(j),UAV(i),SPath(j,i))\right] \end{aligned} \tag{8}$$

The UAV that has the largest degree of membership in "priority of helping" is the one that will be assigned to provide support. The PH fuzzy decision tree is depicted in Figure 2.

4. GP CREATION OF THE AUP TREE

This section will emphasize using the GP as a data mining function to automatically create the AUP tree of the UAVRM. It discusses the terminal set, the function set, the database to be data mined, etc.

4.1 GP's terminal set and function set for creating AUP

The terminal set used to evolve the AUP FDT consisted of the root concepts from the AUP tree and their complements, i.e., the terminal set T is given by

T={RISK-TOL, VALUE, FAST, LOW-RISK, SR, NSR, MP, NOT-RISK-TOL, NOT-VALUABLE, NOT-FAST, NOT-LOW-RISK, NOT-SR, NOT-NSR, NOT-MP}. (9)

Let the corresponding fuzzy membership functions be denoted as

$$\left\{\mu_{risk-tol}, \mu_{value}, \mu_{fast}, \mu_{low-risk}, \mu_{sr}, \mu_{nsr}, \mu_{MP}, \mu_{not-risk-tol}, \mu_{not-valuable}, \mu_{not-fast}, \mu_{not-low-risk}, \mu_{not-sr}, \mu_{not-nsr}, \mu_{not-MP}\right\} \quad (10)$$

By including in the terminal set a terminal and its complement, e.g., "RISK-TOL," and "NOT-RISK-TOL"; "VALUE" and "NOT-VALUABLE"; etc., it is found that bloat is less and convergence of the GP is accelerated. In genetic programming, bloat refers to excessive tree growth. It has been observed that the trees in the evolving GP population will grow by a factor of three every 50 generations[10]. Inclusion of a terminal and its complement is a recent innovation which was not used when FDTs of the earlier RM, the EARM[1], were evolved using a GP. Additional bloat control procedures are described below.

The mathematical form of the complement whether it appears in the terminal set or is prefixed with a "NOT" logical modifier from the function set is one minus the membership function. To make this more explicit

$$\mu_{NOT(A)} = \mu_{not-A} = 1 - \mu_A \quad (11)$$

where *NOT(A)* refers to the application of the logical modifier *NOT* from the function set to the fuzzy concept *A* from the terminal set. The notation, *not-A* refers to the terminal which is the complement of the terminal *A*.

The function set, denoted as F, consists of

$$\text{F}=\{\text{AND1, OR1, AND2, OR2, NOT, POW}_q\} \quad (12)$$

where the elements of (12) are defined below.

Let A and B represent fuzzy membership functions and *q* be a real number, then elements of the function set are defined as

$$AND1(A,B) = MIN(A,B); \quad (13)$$

$$OR1(A,B) = MAX(A,B); \quad (14)$$

$$AND2(A,B) = A \cdot B; \quad (15)$$

$$OR2(A,B) = A + B - A \cdot B; \quad (16)$$

and

$$NOT(A) = 1 - A; \quad (17)$$

$$POW_q(A) = A^q. \quad (18)$$

4.2 Database to be data mined

The database to be data mined is a scenario database like that described for the evolution of the IPDT in EARM[1]. In this instance scenarios are characterized by values of the fuzzy membership functions for the elements of the terminal set plus a number from zero to one indicating the expert's opinion about the value of the fuzzy membership function for AUP for that scenario.

4.3 Fitness function
The input-output fitness for mining the scenario database takes the form

$$f_{IO}(i, n_{db}) \equiv \frac{1}{1 + 2 \cdot \sum_{j=1}^{n_{db}} \left| \mu_{gp}\left(i, e_j\right) - \mu_{\exp ert}\left(e_j\right) \right|}. \tag{19}$$

where e_j is the j^{th} element of the database; n_{db} is the number of elements in the database; $\mu_{gp}(e_j)$ is the output of the fuzzy decision tree created by the GP for the i^{th} element of the population for database element e_j; and $\mu_{expert}(e_j)$ is an expert's estimate as to what the fuzzy decision tree should yield as output for database element e_j.

The AUP tree is evolved in three steps. The first step involves evolving the VMR subtree; the second step, the RMP subtree and the final step, the full AUP tree. In the second and third steps, i.e., evolving the RMP subtree and full AUP tree from the RMP and VMR subtrees, only the input-output (IO) fitness in (19) is calculated.

When evolving the VMR subtree a rule-fitness is calculated for each candidate solution. Only when the candidate's rule fitness is sufficiently high is its input-output fitness calculated. The use of the rule-fitness helps guide the GP toward a solution that will be consistent with expert rules. Also the use of the rule fitness reduces the number of times the IO fitness is calculated shortening the run time of the GP. Finally, the fuzzy rules guide the evolutionary process helping produce FDTs that are closer to human intuition in their geometry. Thus the fuzzy rules also provide a useful bloat control mechanism.

4.4 Rule fitness and fuzzy rules to accelerate convergence
After some preliminary definitions of crisp and fuzzy relations, a set of crisp and fuzzy rules that were used to help accelerate the GP's creation of the VMR subtree are given.

Let T be a fuzzy decision tree that represents a version of the VMR subtree, that is to be evolved by a genetic program (GP). Let A and B be fuzzy concepts. Then let $\gamma_{share}(T, A, B) = 1$ if A and B share a logical connective denoted as C and $\gamma_{share}(T, A, B) = 0$, otherwise.

Furthermore, define the fuzzy relation

$$\mu_{com}(T, A, B, C) = \begin{cases} 0.4 & if \quad C = AND1 \quad or \quad AND2 \\ 0.1 & if \quad C = OR1 \quad or \quad OR2 \\ 0, & otherwise \end{cases} \tag{20}$$

The following rules were used to accelerate the GP's convergence and to help produce a result consistent with human expertise.

R1. "NOT-VALUABLE" and "RISK-TOL" must share a logical connective, denoted as C_1, i.e., it is desired that $\gamma_{share}(T, not-valuable, risk-tol) = 1$

R2. "NOT-VALUABLE" and "RISK-TOL" strongly influence each other, so they should be connected by AND1 or AND2. So it is desired that $\mu_{com}(T, not-valuable, risk-tol, C_1) = .4$

R3. "FAST" and "LOW-RISK" have an affinity for each other. They should share a logical connective, denoted as C_2,

i.e., it is desired that $\gamma_{share}(T, fast, low-risk) = 1$

R4. The fuzzy root concepts "FAST" and "LOW-RISK" strongly influence each other, so they should be connected by AND1 or AND2. So it is desired that $\mu_{com}(T, fast, low-risk, C_2) = .4$.

R5. There is an affinity between the fuzzy root concepts $C_1(not-valuable, risk-tol)$ and $C_2(fast, low-risk)$, they are connected by a logical connective denoted as C_3, i.e., it is desired that,
$\gamma_{share}(T, C_1(not-valuable, risk-tol), C_2(fast, low-risk)) = 1$

R6. The fuzzy composite concepts $C_1(not-valuable, risk-tol)$ and $C_2(fast, low-risk)$ strongly influence each other so it is desired that

$$\mu_{com}(T, C_1(not-valuable, risk-tol), C_2(fast, low-risk), C_3) = .4$$

R7. The elements of *D={not-valuable, risk-tol, fast, low-risk}* should appear on the tree T at least once, i.e.,

$$\mu_{4C}(T) = \begin{cases} 1 & if \quad D's \quad elements \quad present \\ 0 & otherwise \end{cases}$$

R8. The elements of *D* should probably appear only once.

$$\mu_{4C1T}(T) = \begin{cases} .6, & if \quad appear \quad only \quad once \\ .2, & if \quad any \quad appear \quad twice \\ .1, & if \quad any \quad appear \quad \geq 2 \quad times \\ 0, & otherwise \end{cases}$$

The rule-fitness (RF), denoted as $\mu_{RF}(T)$ is defined to be

$$\begin{aligned} \mu_{RF}(T) \equiv \frac{1}{8} \cdot [&(\gamma_{share}(T, not-valuable, risk-tol) + \mu_{com}(T, not-valuable, risk-tol, C_1) + \\ &\gamma_{share}(T, fast, low-risk) + \mu_{com}(T, fast, low-risk, C_2) + \\ &\gamma_{share}(T, C_1(not-valuable, risk-tol), C_2(fast, low-risk)) + \mu_{com}(T, C_1(not-valuable, risk-tol), \\ &C_2(fast, low-risk), C_3) + \mu_{4C}(T) + \mu_{4C1T}(T)] . \end{aligned} \tag{21}$$

For VMR the overall fitness for the i^{th} tree T_i in the evolving population is denoted as $f_{OF}(T_i, n_{db})$. It takes the following mathematical form:

$$f_{OF}(T_i, n_{db}) \equiv \mu_{RF}(T_i) + \chi(\mu_{RF}(T_i) - \tau_{RF}) \cdot f_{IO}(T_i, n_{db}). \tag{22}$$

If $\mu_{RF}(T_i)$ exceeds the threshold, τ_{RF} , then and only then is $f_{IO}(T_i, n_{db})$ added to $\mu_{RF}(T_i)$ yielding the overall fitness; otherwise, the overall fitness is equal to the rule-fitness. To save CPU time the overall fitness is only evaluated if the rule-fitness exceeds the threshold τ_{RF} .

4.5 Tournament selection

The GP program uses tournament selection[10] to accelerate convergence and as one method of dealing with bloat control. In tournament selection, the population is partitioned into tournament subpopulations (TPs). For each TP, the subset of

maximum fitness chromosomes (SMFC) is found. If the SMFC has one element then that chromosome is the winner of the tournament for that TP. If SMFC has more than one element then the subset of minimum depth chromosomes (SMDC) is selected from SMFC. If SMDC has only one element then it is the winner of the tournament for that TP, otherwise a chromosome is selected from the SMDC at random to be the TP's winner.

4.6 Computer algebra

In the preceding subsections bloat has been controlled using adhoc procedures based on tree depth, fuzzy rules embedded in the GP and inclusion of fuzzy concepts and their negations within the GP's terminal set. Most of the bloat in evolving mathematical expressions with a GP arises from the expressions not being in algebraic simplest form[10]. With that observation in mind computer algebra routines have been introduced that allow the GP to simplify expressions. The following is a partial list of algebraic simplification techniques used during the evolution of the IPDT, the AUP, and the PH trees. The simplification routines used when evolving AUP and PH are more sophisticated than those employed for the creation of the IPDT.

One routine simplifies expressions of the form N*OT(NOT(A)) = A*. This can be more complicated than it initially appears, since the NOT logical modifiers can be separated on the fuzzy decision tree.

Another simplification procedure consists of eliminating redundant terminals connected by an AND1 logical connective. An example of this is *AND1(A,A) =A.*

Like the case with the logical modifier NOT there can be a separation between the AND1s and the terminals that add complexity to the simplification operation.

The third algebraic simplification example is like the second. It involves simplifying terminals connected by OR1s. Like AND1, separation between terminals and OR1 can increase the complexity of the operation.

5. GP CREATION OF THE PH TREE

Like the RMP, VMR and AUP fuzzy decision trees both the FBS and PH fuzzy decision trees have been rediscovered using GP based data mining. Like RMP, fuzzy rules were not used to guide the evolution of the FBS and PH trees. So the overall fitness function for both the evolution of FBS and PH is simply the input-output fitness function given by (19).

The terminal and function sets used for the evolution of FBS are given below in (23) and (24), respectively.

T={ VALUE, FUEL, BATTERY, SLOW, RISK-TOL, MP, MR, NSEP, FB, NOT-VALUE, NOT-FUEL, NOT-BATTERY, NOT-RISK-TOL, NOT-MP, NOT-MR, NOT-NSEP, NOT-FB}. (23)

F={MIN3, MAX3, AND3, OR3, MIN, MAX, AND2, OR2, NOT} (24)

The function set elements "MIN" "MAX" and "NOT" are the conventional functions previously used on the AUP tree in Figure 1. The function set elements with "3" in their name are defined in (25-28). The quantities A, B, and C are considered to be positive real number.

$$MIN3(A,B,C) \equiv MIN(MIN(A,B),C); \tag{25}$$

$$MAX3(A,B,C) \equiv MAX(MAX(A,B),C); \tag{26}$$

$$AND3(A,B,C) \equiv A \cdot B \cdot C; \tag{27}$$

$$OR3(A,B,C) \equiv A + B + C - A \cdot B - A \cdot C - B \cdot C + A \cdot B \cdot C \,. \tag{28}$$

There are logical connectives in the function set that do not appear on the FBS tree in Figure 2. Likewise, there are fuzzy concepts in the terminal set that do not appear on the FBS FDT. This is done deliberately to permit the GP to construct a tree that would ultimately be superior to one based on human expertise. Ultimately, the GP finds the tree originally obtained. This relates to the database used. The database is also constructed using human expertise. It is possible that different experts might have made different assignments to scenarios resulting in some of the extra entries in the terminal and function sets being part of the final optimal tree. This has not occurred to date.

Once the FBS fuzzy decision tree was evolved, FBS could be used as a terminal so that GP based data mining could evolve the full PH fuzzy decision tree. The terminal and function set used by the GP to create PH are given below in (29) and (30), respectively.

T={FUEL, BAT, FAST, NSEP, FB, AUP, FBS**,** NOT-FUEL, NOT-BAT, NOT- FAST, NOT-NSEP, NOT-FB, NOT-AUP, NOT-FBS **}**; (29)

F={MIN, MAX, AND2, OR2, NOT, POW_q}; (30)

Like the terminal and function sets for FBS, those for PH contain elements not found on the full PH fuzzy decision tree displayed in Figure 2. These extra elements were added deliberately to allow the GP to evolve a tree different from and superior to the one obtained from expertise, i.e., the FDT displayed in Figure 2. Ultimately, it is the tree in Figure 2 that is rediscovered by the data mining process. Again, the tree ultimately evolved reflects the expert labeled database. Different experts offering different scenarios and output values might well yield an altered form for the PH tree. This has not happened yet. The tree structure and labeling scheme observed in Figure 2 has been reproduced by the GP many times.

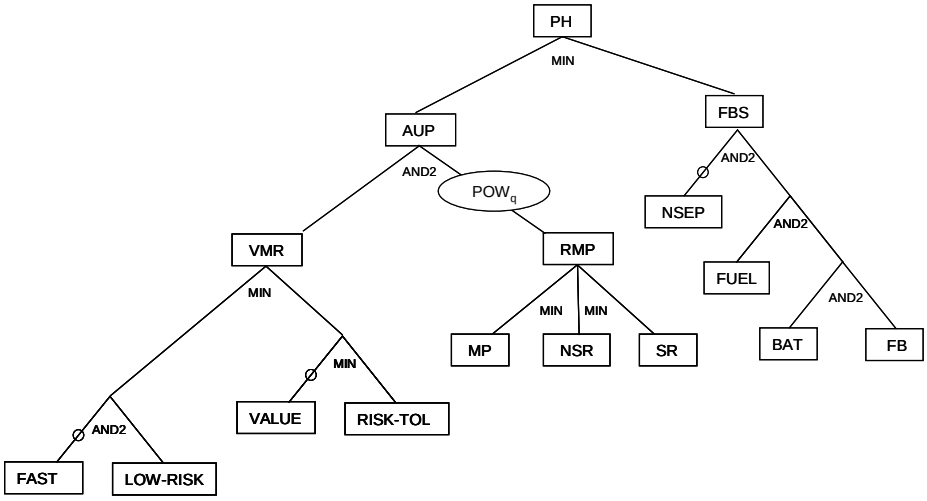

Figure 2: PH tree evolved by GP based data mining.

6. CO-EVOLUTIONARY DATA MINING

Fundamental to the data mining procedure described above are the scenarios that make up the database. There always remains the question, have all the significant scenarios been included in the database? In particular are there scenarios that will cause the planning or control logic to fail?

A co-evolutionary procedure for dealing with this problem has been implemented. Co-evolution in biology refers to the

fact that a biological system never evolves alone, both system and environment simultaneously evolve[11].

In this application of co-evolution a genetic program is used to evolve control logic by mining a database of scenarios. This is followed by using a genetic algorithm (GA) to evolve scenarios that make the control logic fail. Then the pathological scenarios, i.e., the scenarios that make the control logic fail are placed in the database for the GP to DM again resulting in improved logic. This process is iterated a number of times with testing conducted in between with the hopes of ultimately formulating very robust control logic. Fundamental to this process is formulating criteria for what failure means so that the fitness function for the GA can be constructed.

The first quantity related to failure that will be considered is mission risk. It is possible for AUP to assign a UAV to a path with a mission risk that might be considered to be too high by a human expert. On the other hand, a very high mission priority might mandate that this low risk tolerance UAV be assigned to the high risk path. So it is important to combine various concepts to formulate failure criteria. The quantities used to construct failure criteria in this initial effort were the fuzzy concepts VMR, MP, MR and mission completeness denoted as MC. The exact formulation of MC and the fitness function for the GA will be the subject of a future publication.

The GP-GA co-evolutionary process suggests several changes to the PH tree. Let the old fuzzy concept of "priority of helping" be denoted as OPH and the related fuzzy membership function represented by $\mu_{OPH}(UAV(j),UAV(i),SPath(j,i),q)$ where UAV(i) is the UAV requesting help, UAV(j) is evaluating its priority for providing help, $SPath(j,i)$ is the support path that UAV(j) would fly if it helps UAV(i) and "q" is the index and power for the fuzzy modifier POW_q. The GP-GA co-evolutionary data mining process (CEDMP) has suggested many changes to the PH tree, two of which are given below:

1. The potential supporter should have a priority of helping above a certain threshold denoted as τ_{ph1}. So there should be a multiplying Heaviside step function $\chi\lfloor\mu_{OPH}(UAV(j),UAV(i),SPath(j,i),q)-\tau_{ph1}\rfloor$.
2. The value of "q" originally used in AUP and PH was q=2, co-evolution shows that it should be about 1.5.

Given the above modifications, the new priority of help denoted as $\mu_{NEW-PH}(UAV(j),UAV(i),SPath(j,i),q)$ is given in (31) below:

$$\mu_{NEW-PH}(UAV(j),UAV(i),SPath(j,i),1.5) \equiv \mu_{OPH}(UAV(j),UAV(i),SPath(j,i),1.5)\cdot \chi\left[\mu_{OPH}(UAV(j),UAV(i),SPath(j,i),1.5)-\tau_{ph1}\right] \quad (31)$$

7. COMPUTATIONAL EXPERIMENTS

The AUP and PH fuzzy decision trees have been the subject of many experimental tests and very successful in producing expected results[2,3]. The test described in the literature were actually conducted using the AUP and PH fuzzy decision rules as opposed to the GP evolved fuzzy decision trees although the same test were successfully passed by the trees. The FDTs have properties that allow them to be successful when dealing with scenarios that would result in the older decision rules failing. Finally, both FDTs have shown excellent performance in all experiments conducted to date.

8. SUMMARY

A genetic program (GP) has been used as a data mining (DM) function to automatically create decision logic for two different resource managers (RMs). The most recent of the RMs, referred to as the UAVRM is the topic of this paper. It automatically controls a group of unmanned aerial vehicles (UAVs) that are cooperatively making atmospheric measurements.

The DM procedure that uses a GP as a data mining function to create two subtrees of UAVRM is discussed. The resulting decision logic for the RM is rendered in the form of fuzzy decision trees. The fitness function, bloat control

methods, data base, etc., for the trees to be evolved are described. Innovative bloat control methods using computer algebra based simplification are given as well as bloat control procedures based on fuzzy rules embedded in the GP, and the inclusion of terminals and their complements in the terminal set. A subset of the fuzzy rules used by the GP to help accelerate convergence of the GP and improve the quality of the results is provided. A new co-evolutionary approach to data mining is introduced. This approach uses a GP to data mine a data base of scenarios to create an optimal fuzzy decision tree. Afterwards, a genetic algorithm is used to search for pathological scenarios that cause the GP data mined logic to fail. Once these pathological scenarios are found they are incorporated in the GP's database allowing the GP to determine a more robust form of logic. Experimental methods of validating the evolved decision logic are referenced to support the effectiveness of the data mined results.

ACKNOWLEDGEMENTS

This work was sponsored by the Office of Naval Research. The authors would also like to acknowledge Mr. Alan Schultz, Dr. Lawrence Schuette, Dr. Jeffrey Heyer, Dr. Francis Klemm, and Dr. Gregory Cowart.

REFERENCES

1. James F. Smith, III, "Fuzzy logic resource manager: decision tree topology, combined admissible regions and the self-morphing property", *Signal Processing, Sensor Fusion, and Target Recognition XII*, I. Kadar, Vol. 5096, pp. 104-114, SPIE Proceedings, Orlando, 2003.
2. James F. Smith, III and ThanhVu H. Nguyen "Distributed autonomous systems: resource management, planning, and control algorithms", *Signal Processing, Sensor Fusion, and Target Recognition XIV*, I. Kadar, Vol. 5809, pp. 65-76, SPIE Proceedings, Orlando, 2005.
3. J. F. Smith, III and T. H. Nguyen, "Resource manager for an autonomous coordinated team of UAVs," *Signal Processing, Sensor Fusion, and Target Recognition XV*, I. Kadar, Vol. 6235, pp. 104-114, SPIE Proceedings, Orlando, 2006.
4. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Chapter 11, Artech House, Boston, 1999.
5. L.H. Tsoukalas and R.E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, Chapter 5, John Wiley and Sons, New York, 1997.
6. H. J. Zimmerman, *Fuzzy Set Theory and its Applications*. Kluwer, Academic Publishers Group, Boston 1991.
7. J.F. Smith, III and R. Rhyne, II, "A Resource Manager for Distributed Resources: Fuzzy Decision Trees and Genetic Optimization," *Proceeding of the International Conference on Artificial Intelligence, IC-AI'99*, H. Arabnia, Vol. II., pp. 669-675, CSREA Press, Las Vegas,1999.
8. J.P. Bigus, *Data Mining with Neural Nets*, Chapter 1, McGraw-Hill, New York, 1996.
9. J.R., Koza, F.H. Bennett III, D. Andre, and M.A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*. Chapter 2, Morgan Kaufmann Publishers, San Francisco, 1999.
10. S, Luke and L. Panait, "Fighting Bloat With Nonparametric Parsimony Pressure", *Parallel Problem Solving from Nature - PPSN VII. 7th International Conference Proceedings*, J.J.M. Guervos, LNCS Vol.2439, pp. 411-421, Springer-Verlag, Berlin, 2002.
11. D. Cliff and G. F. Miller, "Co-evolution of Pursuit and Evasion II: Simulation Methods and Results," *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96),* P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson (eds.),pp. 1-10, MIT Press Bradford Books, Cambridge, 1996.